

08-11-05

ATT 26 18
126 18



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No. : 09/680,991
Applicant : Murata
Filed : October 6, 2000
Title : STREAMING SERVER ADMINISTRATION PROTOCOL
Group Art Unit : 2126
Examiner : Truong Lechi
Docket No. : 001580-712

BRIEF ON APPEAL

Real Party in Interest

The real party of interest is Apple Computer, Inc.

Related Appeals and Interferences

There are no related appeals, interferences or judicial proceedings.

Status of Claims

Claims 1-3 have been cancelled. Claims 4-20 are pending and have been rejected. Claims 4-7, 11-14 and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,327,608 (Dilligham). Claims 8, 9, and 15-16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Dilligham in view of U.S. Patent 5,999,943 (Nori, et al.). Claims 10, 17, 19 and 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Dilligham in view of U.S. Patent 6,622,170 (Harrison, et al.). All currently pending claims are the subject of this appeal. The claims on appeal are reproduced in the Claims Appendix.

08/12/2005 CNGLYEN2 00000053 09680991
01 FC:1402
500.00 OP

Status of Amendments

No amendments were filed after the final rejection.

Summary of the Claimed Subject Matter

The present invention is directed to a protocol for the retrieval of internal, administrative data from a web server using standard HTTP URLs, which allows the retrieval of this information using standard network software programs, such as a web browser. (Specification, page 3, lines 13-18). The client requests specify the identity of a container within the server that contains administrative data about the server (Specification, page 6, lines 18-21). The requests are processed by an administrative module within the server, which verifies that the request contains the proper container identifier, and responds to the request by transmitting the administrative data to the client. (Specification, page 5, line 20 – page 6, line 6 and Fig. 2).

Issues

1. Whether Claims 4-7, 11-14 and 18 are patentable under 35 U.S.C. § 103(a) over U.S. Patent 6,327,608 to Dilligham.
2. Whether Claims 8, 9, 15 and 16 are patentable under 35 U.S.C. § 103(a) over U.S. Patent 6,327,608 to Dilligham in view of U.S. Patent 5,999,943 to Nori, et al.
3. Whether Claims 10, 17, 19 and 20 are patentable under 35 U.S.C. § 103(a) over U.S. Patent 6,327,608 to Dillingham in view of U.S. Patent 6,622,170 to Harrison, et al.

Argument

I. Claims 4-7, 11-14 and 18 are patentable over Dillingham

There are two problems with the Examiner's rejection of the claims in light of Dillingham. First, the Applicants disagree with the Examiner's assertion that certain elements of the claims are disclosed by Dillingham. Second, there is no motivation at the time of the invention for one of skill in the art to modify Dillingham in the manner contemplated by the Examiner.

In general, Dillingham is directed to the ability to configure web site directories from a remote client, while the present invention claims a method and program for retrieving administrative data from a server utilizing commonly available web based utilities, such as a browser program.

Representative Claim 4 has been reproduced here with the elements that are not disclosed by Dillingham underlined:

4. A method for obtaining internal server data from a computer network having a client and a server, the method comprising the following steps:

generating at the client an HTTP path name having an identity of a container within the server that contains administrative data about the server;

transmitting the HTTP path name from the client to the server;

determining at the server whether the HTTP path name includes the identity of the container of the server;

processing the HTTP path name to retrieve the administrative data from the server if the HTTP path name includes the identity of the container; and transmitting the administrative data back to the client.

In the last office action, with respect to claim 4, the Examiner states that Dillingham teaches an HTTP path name at column 6, lines 50-53, which reads as follows:

... the commands described below for passing path names to the server and returning client scripts and data objects to the client are all securely exchanged over the internet and through the firewall.

While this portion of Dillingham does describe HTTP path names, as are well known in the art, the path names as specified in Claim 4 are HTTP path names *having an identity of a container within the server that contains administrative data about the server*. The HTTP path names disclosed in Dillingham are normal HTTP path names and do not contain the identity of a container containing administrative data about the server therein. Therefore, this limitation is not met by Dillingham.

The Examiner further states that an identity of a container within the server is disclosed in Dillingham at column 2, lines 47-50 and in column 7, lines 55-56. At column 2, lines 47 through 50, Dillingham states as follows:

The server receives the client request and invokes a file system object used to interface the file system. The file system object enumerates the files and/or folders for the directory path specified in the client request.

(emphasis added), and, at column 7, lines 55-60, Dillingham states the following:

Recognizing the path query as pertaining to the file system, the script 110 invokes the file system object 114 and derives a new object based upon the posted path (step 218).

(emphasis added). These portions of Dillingham do not teach that an HTTP path name is *generated at the client* that includes an identity of a container within the server. Instead, these portions of Dillingham teach that once the server receives a client request, *the server* enumerates the files and folders for the path specified in the client request. Claim 4 contains the limitation that the HTTP path name have an identity of a container within the server that contains administrative data about the server. Thus, as recited in claim 4, it is the client, not the server, that specifically identifies a container within the server containing the required administrative data of interest. In Dillingham, the client does not know about the existence of the containers within the server and therefore sends a request to the server asking to be informed of which containers are available. Thus, the limitation that an HTTP path name having an identity of a container within the server that contains administrative data about the server being generated at the client is not taught by Dillingham.

Claim 4 also contains the step of *determining at the server whether the HTTP path name includes the identity of the container of the server*. Because Dillingham does not teach an HTTP path name having an identity of a container being generated at the client and sent to the server, it follows that the step of determining whether the HTTP path name includes the said identity of the container can also not be taught by Dillingham. Rather, in Dillingham, the server recognizes the HTTP request to provide a listing of the files or folders in the path specified by the client request.

Lastly, the step of *processing of an HTTP path name to retrieve the administrative data from the server if the HTTP path name includes the identity of the container* is also not disclosed in Dillingham. For the reasons stated above, the path name from the client in Dillingham does not include the identity of a container. Rather, the request received from the client in Dillingham causes the server to enumerate files and folders for the path specified in the client's request and return that information to the client.

In short, the method claimed in Claim 4 allows the user to retrieve administrative data from the server, while Dillingham discloses a technique that allows a website administrator to remotely browse the directories over the internet. In other words, Dillingham enables the administrator to discover what directories and files exist on the web server.

A further distinction between Dillingham and the claimed invention is that the claimed invention allows the retrieval of administrative data about the server. Examples of the type of data being retrieved are enumerated in the background of the invention portion of the application and include such items as server status, display of the number of current connections, detailed information for each connection, display of configuration options for the server, display of configuration options for each server module extension, etc., data which is not normally available when browsing the files contained in the folder structure of the server. The retrieval of this type of data is not disclosed in Dillingham. Therefore, all portions of the claims that reference the retrieval or transmission of *administrative data* from the server to the client are not taught by Dillingham. Dillingham merely allows for the remote browsing and administration of physical file directories resident on a server from a remote client browser.

The Examiner further states that Dillingham does not teach "generating". The Applicants assume that the Examiner means generating the HTTP path name at the client. The Examiner further states that "Dillingham teach predefined (col 7, ln 6-10)." The Examiner states it would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Dillingham "because Dillingham's predefined would creates the path name requesting from the client to the server". *[sic]* The Applicants assume that the Examiner is referring to the difference noted earlier, that the HTTP path name having the identity of a container within the server that contains administrative data about the server is *generated at the client*. In Dillingham, a request is received and the *server* generates a list of files (pathnames) in response thereto. The problem addressed by Dillingham is that the website administrator does not know what physical files and directories are located on the web server. Without this knowledge, it is impossible to generate at the client an HTTP path name having the identity of the container within the server, and specifically, one that contains administrative data about the server. The Applicants submit that there is no motivation to modify Dillingham in the manner proposed by the Examiner because the approaches taken in Dillingham and the present application with respect to pathnames are different and, in fact, opposite from each other. The path name is unavailable at the client in Dillingham, while, in the present application, the path name is generated at the client. Therefore, one of ordinary skill in the art would not look to Dillingham for guidance regarding how to generate the path name at the client or for a suggestion to do so. Even if the inventors had knowledge of Dillingham at the time of the invention, no teachings in Dillingham suggest generating a path name at the client, rather than making an

inquiry at the server for the path names. As a result, the Applicants submit that there is no basis for the Examiner's purported modification, or interpretation, of Dillingham in the manner shown to render the present invention obvious.

The Applicants respectfully submit that Claim 4 is not obvious and likewise, independent Claims 11 and 18 are not obvious or suggested by Dillingham.

II. Claims 8, 9 and 15-16 are Patentable Over Dillingham in view of Nori

As dependent claims, Claims 8, 9, and 15-16 contain the same limitations as above with respect to Claim 4. Therefore the comments above regarding Dillingham also apply to these claims. The Examiner further cites Nori, et al. for the concept of a snapshot of data. The Examiner states that Dillingham does not teach that the administrative data is a snapshot of a prescribed value.

The application defines a snapshot as follows:

A snapshot is the container of values from which an admin response is generated (the measured values at a particular point in time, as compared to the constantly changing values). There is one active snapshot per admin connection. Each request without a snapshot in the URI generates a new snapshot. There is only one current snapshot per request. Each new snapshot invalidates the previous per admin connection snapshot. *A snapshot is a hierarchy of containers specifying the type, name, value and read/write attribute of values copied from the server's internal structures using the module API (Application Programming Interface).*

(emphasis added). Therefore a snapshot is a hierarchy of containers containing name/value pairs for internal administrative server variables.

The Examiner states that this is disclosed in Nori, et al. at column 6, lines 60-63, and at column 7, lines 25-30.

Nori, et al. teaches a method and system for performing operations on large objects ("LOBs") in a database. Column 6, lines 60-63 of Nori discloses a snapshot-id, which is a field containing a value that indicates consistent read version of an LOB. The snapshot-id is therefore a field which contains a value that indicates whether the LOB has been read correctly from the database, not a hierarchy of containers. Column 7, lines 25-30 indicate that a snapshot is a version of the database which reflects all changes committed to the database as of a particular point in time and that no change is committed after that particular point in time. Because of the differences between the claimed invention and Nori, the Applicants submit that Nori does not disclose the snapshot as that term is used in the present invention.

For a claim to be obvious from a prior art reference or combination of references, the reference or references must disclose each element of the claim and contain a teaching, suggestion or motivation to combine the references in manner to create the claimed invention. In re Sernaker, 702 F.2d 989, 995-996; 217 U.S.P.Q. 1, 6 (Fed. Cir. 1983). It is wrong to use applicant's disclosure "as a guide through the maze of prior art references, combining the right references in the right way so as to achieve the results of [applicants'] claims. Monday morning quarterbacking is quite improper when resolving the question of nonobviousness..." Orthopedic Equipment Co., Inc. et al. v. United States, 702 F.2d 1005, 1012; 217 U.S.P.Q. 193, 199 (Fed. Cir. 1983).

The applicants submit that there is no motivation to combine Dillingham and Nori and that to do so is hindsight construction on the part of the Examiner. Nori deals with large objects in a database, while the present invention deals with containers containing administrative data about a server. Thus, a person of skill in the art at the time of the invention would not have looked to Nori for guidance in handling data in containers. Additionally, there is no teaching in Nori or in Dillingham that suggests or motivates their combination. The references deal with different subject matter; Dillingham with file structures on a server and Nori with large data objects in a database. No teaching exists in either reference to suggest that their combination would lead to the use of snapshots in the present invention, especially because of the differences in the snapshots of Nori and the snapshots of the present invention, as discussed above. The Applicants submit that the combination of these references is an example of “Monday morning quarterbacking” by the Examiner and that, as a result, the combination of the references is improper.

III. Claims 10, 17, 19 and 20 are Patentable Over Dillingham in View of Harrison

As dependent claims, Claims 10, 17 and 19-20 contain the same limitations as above with respect to Claim 4. Therefore the comments above regarding Dillingham also apply to these claims. In addition, the Examiner states that Harrison, et al. teaches changing the value of the server with an HTTP path name. The Applicants submit that the Examiner has misread Harrison. Harrison provides a facility for the modification of lightweight directory access protocol (LDAP) client databases. A directory, as the term is used in Harrison, is a specialized database, also

called a data repository, that stores typed and ordered information about objects. (See Harrison, column 1, lines 25-27). Thus, the object that is being changed within Harrison is a file contained on a server, namely, a database-type file. Harrison does not mention reading or changing the internal variables of a server (i.e., the administrative data). Further, the database (not internal administrative data) in Harrison is not being modified via an HTTP path name. The portion of Harrison cited by the Examiner (column 6, lines 44-46) states that a “VPN configuration application (15) sets the path of information to allow updates to the policies while retrievals are occurring.” Here, it is a VPN (“virtual private network”) configuration application which is initiating a request to change a database. No mention is made in Harrison of using a browser or an HTTP path name to provide a request to change the internal policy parameters of a server. In fact, Harrison specifically mentions that SNMP (Simple Network Management Protocol) messages are used to set the path of information flow. (See Figs. 1 & 4, col. 6, lines 49-51). Therefore, the Applicants submit that the combination of Harrison and Dillingham do not, in combination, suggest or teach all elements of the present invention.

Further, the Applicants submit that there is no motivation to combine Dillingham and Harrison. Dillingham deals with file structures on a server, while Harrison deals with LDAP databases. Their combination is not suggested or motivated by either reference. Further, because the present invention deals with enhanced HTTP URLs and Harrison deals with SNMP messages, a person of skill in the art at the time of the invention would not look to Harrison for guidance in retrieving data with HTTP paths. Lastly, the problems being solved in the present invention and Harrison are different. In the present invention, the inventors are solving the

problem of retrieving data from an HTTP server without utilizing extensive resources and enabling the retrieval of that data with tools that are readily available (i.e., a web browser). Harrison, addresses the problem of allowing access to information by clients while other clients are attempting to update the data (See Harrison, column 3, lines 29-32). Because the present invention is not concerned with this problem, no motivation exists to refer to the teachings of Harrison, either alone or in combination with Dillingham. As a result, the Applicants submit that the application of Harrison is hindsight construction and that the combination of Dillingham and Harrison to suggest the present invention is improper.

Conclusion

There is no teaching or suggestion in the cited references, alone or in combination, of an HTTP protocol that allows the reading and modification of internal administrative data parameters from a container on a network server, as claimed in currently pending Claims 4-20. All elements of the claims are simply not taught by the combination of references, nor is there sufficient motivation to combine Dillingham with Nori or Harrison. Therefore, the reversal of the rejections of the claims on appeal is respectfully requested.

Respectfully submitted,



Dennis M. Carleton
Registration No. 40,938
BUCHANAN INGERSOLL, P.C.
One Oxford Centre
301 Grant Street, 20th Floor
Pittsburgh, PA 15219-1410
(412) 562-1895

Attorney for Applicant

Claims Appendix

4. A method for obtaining internal server data from a computer network having a client and a server, the method comprising the following steps:

generating at the client an HTTP path name having an identity of a container within the server that contains administrative data about the server;

transmitting the HTTP path name from the client to the server;

determining at the server whether the HTTP path name includes the identity of the container of the server;

processing the HTTP path name to retrieve the administrative data from the server if the HTTP path name includes the identity of the container; and

transmitting the administrative data to the client.

5. The method of claim 4 wherein an administrative module of the server determines whether the HTTP path name includes the identity of the container.

6. The method of claim 4 wherein an administrative module of the server processes the HTTP path name to retrieve the administrative data.

7. The method of claim 4 wherein the server transmits an HTML page if a container is not identified within the HTTP path name.

8. The method of claim 4 wherein the administrative data is a snapshot of prescribed values.
9. The method of claim 8 wherein the snapshot of prescribed values is generated at a particular point in time.
10. The method of claim 4 further comprising the step of changing a value of the server with the HTTP path name.
11. A computer readable medium containing a program which implements the following procedure:
 - generating at a client an HTTP path name having an identity of a container within a server that contains administrative data about the server;
 - transmitting the HTTP path name from the client to the server;
 - determining at the server whether the HTTP path name includes the identity of the container of the server;
 - processing the HTTP path name to retrieve the administrative data from the server if the HTTP path name includes the identity of the container; and
 - transmitting the data to the client.
12. The computer readable medium of claim 11 wherein an administrative module of the server determines whether the HTTP path name includes the identity of the container.

13. The computer readable medium of claim 11 wherein an administrative module of the server processes the HTTP path name to retrieve the administrative data.

14. The computer readable medium of claim 11 wherein the server transmits an HTML page if a container is not identified within the HTTP path name.

15. The computer readable medium of claim 11 wherein the administrative data is a snapshot of prescribed values.

16. The computer readable medium of claim 15 wherein the snapshot of prescribed values is generated at a particular point in time.

17. The computer readable medium of claim 11 further comprising the step of changing a value of the server with the HTTP path name.

18. A computer network comprising:

 a client computer configured to generate an HTTP path name that identifies a container within the server that has administrative data about the server.

 a server computer in communication with the client computer, the server computer configured to determine whether the HTTP path name includes the identity of the container and

to process the HTTP path name to retrieve the administrative data from the server if the HTTP path name includes the identity of the container, the server computer being configured to transmit the administrative data to the client computer.

19. The network of claim 18 wherein the client computer is configured to transmit and HTTP path name that changes a value of the server computer.

20. The network of claim 19 wherein the server computer is configured to change a value of the server computer in response to the HTTP path name transmitted from the client computer.

Evidence Appendix

No additional evidence is being submitted.

Related Proceedings Appendix

There are no related proceedings.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



Appl. No. : 09/680,991
Applicant : Murata
Filed : October 6, 2000
Title : STREAMING SERVER ADMINISTRATION PROTOCOL
Group Art Unit : 2126
Examiner : Truong Lechi
Docket No. : 001580-712

**PETITION FOR ONE (1) MONTH EXTENSION
OF TIME UNDER 37 C.F.R. § 1.136**

Applicants hereby petitions for a one (1) month extension of time, pursuant to 37 C.F.R. § 1.136, to extend the period for response regarding the Brief on Appeal in the above-identified application up to and including August 11, 2005. A check in the amount of \$120.00 is enclosed as the requisite fee under 37 C.F.R. § 1.17(a)(2).

The Commissioner is hereby authorized to charge any underpayment or credit any overpayment associated with this communication to Deposit Account No. 02-4553. A duplicate copy of this sheet is attached.

Respectfully submitted,

Dennis M. Carleton
Registration No. 40,938
BUCHANAN INGERSOLL, P.C.
One Oxford Centre
301 Grant Street, 20th Floor
Pittsburgh, PA 15219-1410
(412) 562-1895

Attorney for Applicant

08/12/2005 CNGUYEN2 000000053 09688991

500-00-00
120.00 00

01-FEB-09
02 FC:1251



CERTIFICATE OF MAILING BY "EXPRESS MAIL" UNDER
37 CFR 1.10 - SEPARATE PAPER

ATTORNEY'S DOCKET NO.

001580-712

IN RE APPLICATION OF

Murata

Serial No.

09/680,991

Filed

10-06-2000

FOR

Streaming Server Administration Protocol

GRP. ART UNIT

2126

EXAMINER

Truong Lechi

"Express Mail" mailing label number

EV 662116478 US

Date of deposit:

August 10, 2005

I hereby certify that this paper or fee is being deposited with the
United States Postal Service "Express Mail Post Office to Addressee"
service under 37 CFR 1.10 on the date indicated above and is addressed
to Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-
1450

LouAnne Guerra

(Typed or printed name of person mailing paper or fee)

LouAnne Guerra
(Signature of person mailing paper or fee)